Maintaining Connectivity in UAV Swarm Sensing

Maintaining Connectivity in UAV Swarm Sensing

W. T. Luke Teacy, Jing Nie, Sally McClean and Gerard Parr
School of Computing and Information Engineering
University of Ulster, UK
Email: {l.teacy, j.nie, gp.parr, si.mcclean}@ulster.ac.uk

***Abstract*—In many applications, Unmanned Aerial Vehicles (UAVs) provide an indispensable platform for gathering information about the situation on the ground. However, to maximise information gained about the environment, such platforms require increased autonomy to coordinate the actions of multiple UAVs. This has led to the development of flight planning and coordination algorithms designed to maximise information gain during sensing missions. However, these have so far neglected the need to maintain wireless network connectivity.**

In this paper, we address this limitation by enhancing an existing multi-UAV planning algorithm with two new features that together make a significant contribution to the state-of-the-art: (1) we incorporate an on-line learning procedure that enables UAVs to adapt to the radio propagation characteristics of their environment, and (2) we integrate flight path and network routing decisions, so that modelling uncertainty and the affect of UAV position on network performance is taken into account.

I. Introduction

In many civilian applications, an aerial view is invaluable for gaining information about the situation on the ground [1]. Such applications include wilderness search and rescue, environmental monitoring, and situation awareness in natural disasters. In manned flight, such scenarios place a heavy burden on pilots, requiring long hours of monotonous flight at high-levels of concentration. Increasingly, however, advances in airframe design and control technology mean that using Unmanned Aerial Vehicles (UAVs) for such tasks is becoming a viable option. Small, inexpensive aircraft are now commercially available, and are typically equipped with an array of on-board sensors, such as GPS receivers and gyroscopes for navigation; and visible or infrared cameras to provide real-time information about the environment below. Moreover, with the development of sophisticated flight control algorithms, many craft can now take-off, land and fly automatically. As such, the human operator is no longer required to take low-level control of the UAV, but can instead concentrate on high-level decisions, and navigate the vehicle via GPS way-points.

Despite these developments, existing applications still require a user on the ground to make complex real-time decisions about how to utilise the UAVs while they are in the air. Although such ground-based control has several advantages over manned flight, the complexity of some tasks mean that it is impossible for a human operator to take maximum advantage of UAV resources without increased autonomy or decision support mechanisms. For example, with multiple UAVs, the information gained about the environment is not just the sum of observations made by all UAVs. Rather, the actions of each UAV must be coordinated to reduce redundancy of effort, or to increase the accuracy of estimates by fusing data taken by different UAVs from different viewpoints.

Such requirements have lead to the development of a number of sophisticated path planning algorithms, which coordinate the actions of multiple UAVs to maximise the information gained about the subject of interest [2], [3], e.g. the concentration of airborne pollutants or the location of a missing person [4]. At the same time, advances in mobile network technology have been made that attempt to maintain connectivity between mobile devices as they move within their environment [5]. In the real world, however, it is often not sufficient to consider networking and sensing as separate issues, since both are affected by UAV position, and so both should be used to inform path planning decisions. Moreover, although some progress has been made on maximising sensor coverage under communication constraints [6], this does not directly optimise information gain (which is typically the primary objective), nor are environmental effects on network Quality of Service (QoS) usually considered.

In this paper, we address these limitations by introducing a new decision making mechanism in which the need to maximise information gain is balanced against network connectivity and management requirements. More specifically, we focus on applications in which a team of multiple UAVs are dispatched for some sensing task, but must at all times remain in contact with a base station via a short range ad-hoc wireless network. For example, Fig. 1 illustrates one such application, in which a swarm of UAVs must spread out to find a missing person as quickly as possible and before their energy supplies run out. At the same time, however, they must remain within range to maintain multi-hop connectivity with the base station. To achieve this, we adapt an existing mobile sensor path planning algorithm, proposed in [3], by incorporating two new features that together make a significant contribution to the state-of-the-art. First, we modify the algorithm to account for the limited communication range of each UAV. However, rather than assume that this range is fixed, we accept that it may change in response to the environment and its topography. We therefore incorporate an on-line learning mechanism that allows the UAVs to adapt to the observed radio characteristics at their current positions. Second, to enable UAVs to move out of direct communication range with the ground station, we incorporate a multi-hop routing protocol into the algorithm. This ensures that, as the UAVs decide where to move next, they account for the expected cost of their movements on maintaining communication with the base station.

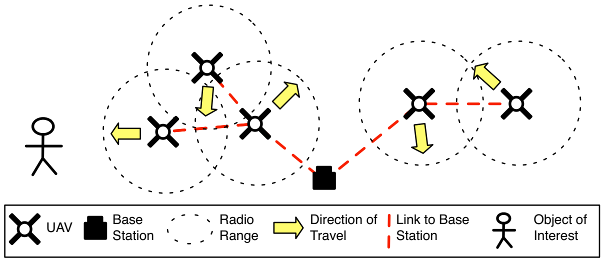

Fig. 1. In this example, 5 UAVs search a designated area to find a missing person as quickly as possible, and before their energy supplies run out. However, for control and data transmission, they must maintain a multihop communication link with the base station.

The rest of this paper is structured as follows: Section II outlines related work on multiagent planning for sensing tasks; Section III introduces our communication learning and modelling framework, and using this framework, shows how an existing distributed algorithm can be adapted for coordination and routing among multi-UAVs in sensing applications; and finally, Section IV concludes and discusses future work.

II. Background

A. Path Planning in multi-UAV Sensing Applications

As discussed in Section I, the primary use envisaged for UAVs in civilian applications is as mobile sensor platforms to gather information about some unknown quantity, θ, such as the location of a missing person. When planning the actions of a group of UAVs, the key objective is therefore to choose paths for each UAV that together maximise the information gained about θ. Using information theory, this goal can be formalized by defining some prior probability distribution, $Pr(\theta)$, and posterior distribution, $Pr(\theta|D)$, where D is the data collected by the UAVs. The objective is then to maximise some information theoretic measure of the reduction in uncertainty encoded in $Pr(\theta|D)$ relative to the prior $Pr(\theta)$. Although a number of suitable measures have been proposed, most are derived from the entropy of the prior and posterior distributions, such as KL divergence and mutual information [7].

In general, using such measures to find the optimal paths for a group of UAVs is a hard combinatorial problem. This is because, in principle, the Cartesian product of all possible paths for each UAV must be considered. Fortunately, under most conditions, entropy based measures usually exhibit two characteristics that aid their optimisation [8]:

1) *Submodularity* — Applied to information theoretic measures, this formalises the intuitive property that more data never decreases information, but the value of new data is less when more prior information is available.
2) *Locality* — This property holds when sensor observations are spatially related. Specifically, observations that are taken close together in space and time contain more shared redundant information than observations made at different locations or at different times.

By taking advantage of these two properties, it is possible to produce efficient algorithms for choosing near optimal locations for taking sensor readings. In particular, when the objective (or *utility*) function is submodular, greedy approaches for choosing sensor locations are guaranteed to gather information within a certain bound of the maximum possible, which is as least as good as any other polynomial time algorithm unless P=NP [9]. Although this result does not apply directly to path planning problems,[1] it has still enabled the development of efficient greedy path planning algorithms with albeit slightly lower guarantees on performance [2].

Although such algorithms provide a computationally efficient means to maximise information gain, their performance is directly tied to the fundamental characteristics of information theoretic utility functions. Unfortunately, these properties break down when communication costs are taken into account, because objective functions for optimising network performance are not guaranteed to exhibit the same properties. On the contrary, making more observations does not guarantee higher reward if the cost of those observations is lost network connectivity (thus violating submodularity); and multi-hop routing means that the costs of one UAV depends on all other (possibly remote) UAVs along its communication path to the ground station (thus breaking locality).

Nevertheless, the problem of jointly optimising network connectivity and information gain still retains some structure that can be harnessed to produce efficient planning algorithms (see Sec. III). The key challenge is then to modify existing planning mechanisms so that they can account for communication concerns without significantly degrading performance. For this purpose, we build on the work of Stranders' *et al* [3] who propose a distributed algorithm for planning the routes of multiple mobile sensors (e.g. UAVs). Their approach takes advantage of the locality property by only considering the impact of a sensor's immediate neighbours on the value of its observations, and also adopts a greedy path pruning strategy (thereby benefiting from submodularity). However the core of their approach is based on the max-sum algorithm, which is a more general tool for optimisation that does not intrinsically rely on these properties. As such, it has the potential for adaption to the more complicated joint communication and sensing problem proposed here, and for this reason, we use it as a basis for the techniques introduced in this paper. Before doing so, however, we now describe the max-sum algorithm in more detail as a prerequisite to understanding the material that follows.

B. Optimisation using the Max-Sum Algorithm

Following [3], we define a decentralized coordination problem as one in which a set of m agents must together choose a *joint* action, $\mathbf{a} \in \mathcal{A}$, with the goal of maximising a single shared utility function $U : \mathcal{A} \to \mathbb{R}$. Here, the agents' joint action is

[1] The optimality guarantees for greedy optimisation of submodular functions only applies when the location for sensor observations is unconstrained. This does not apply to UAV path planning problems, because sensor locations are limited by the speed a UAV can travel between different points [2].

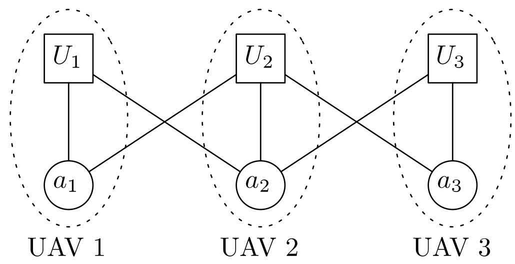

Fig. 2. Example of a factor graph, representing a coordination problem involving three UAVs, which share a utility function $U(\mathbf{a}) = U_1(a_1, a_2) + U_2(a_1, a_2, a_3) + U_3(a_2, a_3)$. Here, each UAV is associated with one factor and action variable, represented by squares and circles respectively.

itself a set of individual actions, $\{a_i\}_{i=1}^m$, where $a_i \in \mathcal{A}_i$ is the action performed by the ith agent, and $\mathcal{A}_i$ is the set of actions it has available. In general, this is a hard combinatorial problem, because the set of possible joint actions ($\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_m$) grows exponentially with the number of agents. However, in many applications, the utility function can be *factored* into a sum of simpler functions (factors) that each depend on only a subset of agent actions:

$$U(\mathbf{a}) = \sum_{j=1}^{u} U_j(\mathbf{a}_j), \quad \text{where } \mathbf{a}_j \subseteq \mathbf{a} \tag{1}$$

The agents' goal is thus to choose $\mathbf{a} = \arg\max_{\mathbf{a}} \sum_j U_j(\mathbf{a}_j)$. This problem can be efficiently solved by the max-sum algorithm, which exploits the limited action dependence of the individual factors to avoid enumerating the entire joint action space. This is achieved in two stages.

First, the problem is represented as a bipartite graph, known as a *factor* graph [10], which comprises two disjoint sets of vertices representing the set of individual agent actions, $\{a_i\}_{i=1}^m$, and the factors of the utility function, $\{U_j\}_{j=1}^u$, respectively. The edges of the graph represent factor dependencies on individual actions. For example, Fig. 2 represents the utility of a group of three cooperating UAVs, each associated with an individual factor and action variable. Although such a one-to-one mapping between factors and agents (or, as in this case, UAVs) is not necessary, decomposing a problem in this way is convenient from an implementation point of view. This is because the max-sum algorithm works by *passing messages* along the *edges* of the factor graph, so by making each UAV responsible for transmitting messages to and from its own factor and action vertices, it is straightforward to distribute the algorithm by sending messages between UAVs over the wireless network. Specifically, in our example, UAVs 1 & 2 coordinate by exchanging messages with UAV 2, but do not need to communicate directly because their corresponding vertices are not directly connected.

Second, no matter how responsibility for message passing is distributed among the agents, the defining property of the max-sum algorithm is the specification of the messages. In particular, for each related factor-action pair (U_j, a_i), messages are passed in both directions along their shared edge, which specifies how the value of the global utility U changes in response to each possible value of a_i. For each value of $a_i \in \mathcal{A}_i$, these are denoted as $q_{i\to j}(a_i)$ for messages passed from a_i to U_j, and $r_{j\to i}(a_i)$ for messages passed from U_j to a_i. These are defined in turn as follows:

From variable to factor: $q_{i\to j}(a_i) = \alpha_{ij} + \sum_{k\in\mathcal{M}_i\setminus j} r_{k\to i}(a_i)$, *where* $\mathcal{M}_i$ *is a vector of factor indices indicating which factor nodes are connected to variable node* i, *and* α_{ij} *is a normalising constant to prevent message values increasing without bound in cyclic graphs.* $\Box$

From factor to variable: $r_{j\to i}(a_i) = \max_{\mathbf{a}_j\setminus i}[U_j(\mathbf{a}_j) + \sum_{k\in\mathcal{N}_j\setminus i} q_{k\to j}(a_k)]$, *where* $\mathcal{N}_j$ *is a vector of variable indices indicating which variable nodes are connected to factor node* j, *and* $\mathbf{a}_j\setminus i \equiv \{a_k : k \in \mathcal{N}_j \setminus i\}$. $\Box$

Given that these messages define a different value for each $a_i \in \mathcal{A}_i$, assuming $\mathcal{A}_i$ is finite,[2] they are typically represented as a vector of size $|\mathcal{A}_i|$, containing one value for each element of $\mathcal{A}_i$. Various schemes are then possible to decide when to start and stop sending messages, but a common and robust strategy is to initialise all messages to $\mathbf{0}$, and then repeatedly update them asynchronously according to their definitions until they converge, or some other stopping condition is reached. The action for each agent is then selected by setting $a_i = \arg\max_{a_i} \sum_{j\in\mathcal{M}_i} r_{j\to i}(a_i)$.

When the factor graph is cycle free, the number of messages each vertex must send to guarantee convergence is equal to twice the depth of the tree. When applied to cyclic graphs, the algorithm is no longer guaranteed to converge to the optimal solution. However, extensive empirical evidence has demonstrated that optimal or near optimal solutions are still generated in practice after a relatively small number of iterations [12]–[14]. Moreover, compared to similar algorithms that have stronger convergence guarantees, the max-sum algorithm scales well to problems involving large numbers of agents. This is because both communication and computation costs depend only on the local interactions of the variables.[3] Thus, the max-sum represents an effective way to balance the need to optimise the coordinated actions of a group of UAVs against the computational and communication costs of doing so.

C. Applying Max-Sum to Mobile Sensor Problems

We now describe the approach proposed by Stranders *et al* [3], which shows how the max-sum algorithm can be used to solve sensing problems involving teams of mobile agents. In Section III-B, we then use this as a basis for solving the more complex problem of coordinating a team of UAVs for sensing problems with communication constraints.

[2]Implementations of max-sum that deal with continuous value actions are also possible, for example, by using piecewise linear functions [11].

[3]In particular, when interactions among variables are only pairwise the computational complexity of the maximisation step performed by each function scales linearly with the number of variables connected to the function, while in general this computation is exponential only in the number of variables connected to the function. Moreover, in all cases, the size of each message depends only on the domain size of the variable.

First, as described in the previous section, each UAV, i, is assigned a local utility function, U_i, and an action variable, $a_i \in \mathcal{A}_i$, which form the vertices of a factor graph on which max-sum operates. Ideally, the action space, $\mathcal{A}_i$, of each UAV should contain all possible flight paths from its current position. However, for simplicity, it is limited to a sequence of N movements in one of 8 equally spaced compass directions over a fixed distance. For example, if $N = 5$ each possible action amounts to a choice of 8^5 possible paths of for each UAV, consisting of 5 discrete movements in 8 possible directions. For longer paths, the algorithm is run periodically to plan the next N steps from the UAVs' current positions — stopping only when the objective is met or energy supplies are depleted. Thus, although the approach is myopic, the total path length is not limited. Moreover, although the original work constrained movements to two dimensions, this can be easily extended to include movement in three dimensions.

With this in mind, each UAV's path results in new observations being made at N positions, resulting in a reduction in entropy (increase in information) about the quantity of interest, θ. The utility function for each UAV is thus given by the reduction in entropy about θ, given the paths $\mathbf{a}_j = \{a_j | j \in \mathcal{N}_i\}$. Here, the problem formulation takes advantage of locality, by only including in $\mathcal{N}_i$ UAVs within a fixed distance of UAV i's location. UAVs outside this radius are assumed to have negligible affect on i's local utility, and can therefore be safely ignored. The result is a factor graph in which a UAV's utility U_i defines a factor that is connected to all the action variables of $\mathcal{N}_i$, including i's own action. For example, the factor graph in Fig. 2 represents a case in which UAVs 1 & 3 are both in range of UAV 2, but not of each other. [4]

Now, despite max-sum efficiency, the size of the individual action spaces can still be a problem when agents have more than a few neighbours. For this reason, Stranders *et al* propose two pruning techniques that further reduce the number of joint actions that must be considered in a neighbourhood. The first operates before max-sum is run, and works by greedily pruning the dominated actions of individual agents — that is, actions that cannot result is optimality no matter what the actions of neighbouring UAVs. The second then employs a branch and bound procedure while max-sum is running, which prunes the neighbourhood joint actions while messages are being calculated. In their experiments, Stranders' *et al* show that these pruning techniques can reduce the joint action space by up to 92%. We now show how this procedure can be further refined to account for network connectivity requirements.

III. Network Aware UAV Sensing

A. Learning and Modelling Connection Costs

To enable our UAVs to make rational decisions about network connectivity constraints, two things are required: (1) a *radio propagation model* to predict the affect that flying to a particular set of positions will have on the quality of service (QoS) of the communication links; and (2) an objective way to assess the value of a communication link, given its QoS properties. Both of these requirements are application dependent, since different QoS metrics may be appropriate depending on the type of data that must be transmitted, and the characteristics of the UAVs' environment that may affect the QoS of the network. Therefore, rather than commit to any particular QoS model, we instead propose a generic framework that can incorporate a variety of different QoS attributes and radio propagation models, depending on the application requirements. We now explain how each of these two requirements are met below.

First, since radio propagation (and therefore network QoS properties) are highly dependent on environmental conditions, various physical models have been proposed to predict how radio waves travel in different situations, such as indoor, urban, or rural environments. However, none of these models can produce perfect predictions in every situation. We therefore adopt an on-line learning approach, in which UAVs use a theoretical model suitable to their environment, but can adapt the model to compensate for any observed deviation between the theoretical predictions and actual channel performance.

More specifically, suppose that $\mathcal{Q}$ is an appropriate application dependent space of QoS attribute vectors, and that we are interested in predicting the QoS $q \in \mathcal{Q}$ at time t of the channel link between UAVs at positions $p_1, p_2 \in \mathbb{R}^3$. Then, given a suitable theoretical model, $f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathcal{Q}$, we assume that the true value q at time t is given by

$$q = f(p_1, p_2) + n(p_1, p_2, t) \tag{2}$$

where n is a time dependent noise distribution, which the UAVs learn from observed channel performance as they fly. For example, in the simplest case, $\mathcal{Q}$ could represent signal loss in dBm, and f the expected signal loss according to the free space model. Any observed discrepancy between this model and actual signal loss (e.g. due to obstacles or multipath propagation) could then be accounted for by adapting n.

Assuming an appropriate choice of f is available, we now show how n may be specified and adapted in response to observed QoS. For this purpose, we define n to be an arbitrary function drawn from a *Gaussian process* ($\mathcal{GP}$), which is a class of non-parametric stochastic models that are commonly used for regression and classification problems (see [15] for details). In our context, this approach has three main advantages: (1) it places no constraints on the shape or form of n, (2) it allows us to learn how n varies with time just as easily as its variance with position, and (3) it provides a position and time dependent measure of the uncertainty in our predictions. This last advantage is of particular importance, because it allows us to adapt UAV behaviour according to the risk associated with more uncertain predictions. For example, it may be appropriate for UAVs to stay closer together when flying in areas where there is greater uncertainty about channel performance, and therefore a higher risk of losing connectivity.

Finally, to provide our UAVs with a means of assessing the relative value of different QoS properties, we must specify a utility or cost function, $L : \mathcal{Q} \to \mathbb{R}$, such that $L(q_1) > L(q_2)$

[4] Note that, due to the way this problem is factored, $\forall i$, $\mathcal{M}_i = \mathcal{N}_i$.

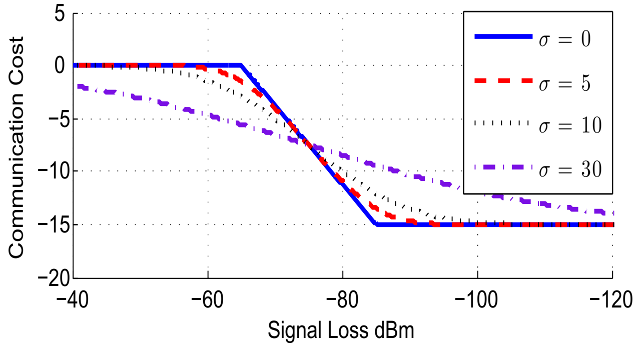

Fig. 3. Expected Link Cost for varying σ

if and only if q_1 is preferred to q_2. As described in Section III-B, this is used to make informed decisions about where to fly next. However, as its precise definition relies on application specific QoS requirements, we do not give a concrete definition, but instead allow L to be any *piecewise linear* function. More specifically, this implies that $\mathcal{Q}$ can be divided into a finite set of disjoint subsets $\{\mathcal{Q}_i\}_{i=1}^n$, such that $\mathcal{Q} = \bigcup_i \mathcal{Q}_i \wedge \forall i \neq j,\ \mathcal{Q}_i \cap \mathcal{Q}_j = \emptyset$, and for any $q_i \in \mathcal{Q}_i$, L has the form $L(q_i) = a_i \cdot q_i + b_i$ where a_i and b_i are constant scalars. The benefit of using piecewise linear functions is that they can approximate any other function with arbitrary precision, but at the same time, maintain a simple form that allow UAVs to make decisions that account for uncertainty in QoS predictions in an efficient and tractable manner.

This is possible due to the general properties of linear functions and our choice representation of n using $\mathcal{GP}$'s. For example, suppose that $\mathcal{Q}$ is divided in subsets as above, where a_i and b_i are the coefficients for the linear function defined over $\mathcal{Q}_i$. According to decision theory, a rationale agent should always act to maximise their expected utility, which in this case, can be calculated by the following closed form equation:

$$EL(p_1, p_2) = \sum_{i=1}^{n} Pr(q \in \mathcal{Q}_i)\left(a_i \cdot E[q|q \in \mathcal{Q}_i] + b_i\right) \quad (3)$$

where $E[q|q \in \mathcal{Q}_i]$ is the expected value of q given the positions p_1, p_2 and that it belongs to $\mathcal{Q}_i$, and $Pr(q \in \mathcal{Q}_i)$ is the corresponding probability. Now, since n is drawn from a $\mathcal{GP}$, we know from the defining property of a Gaussian process that q will have a Gaussian distribution with mean, μ, giving its expected value, and covariance, Σ, encoding the uncertainty about q. With this in mind, $Pr(q \in \mathcal{Q}_i)$ is calculated from the Gaussian cumulative distribution function, and $E[q|q \in \mathcal{Q}_i]$ is the expected value of a truncated Gaussian [16].[5] The result of this calculation is illustrated in Fig. 3 for $\mathcal{Q}$ defined as the (one dimensional) signal loss (dBm), and varying standard deviation $\sigma = \sqrt{\Sigma}$. Here, the solid line represents an initial piecewise linear function, that assigns a near zero cost when signal path loss is < 65, a cost of 15 when the signal path loss is > 85, or some intermediate value when it is between 65 and 85. For example, this could represent an application in which a path loss of less that 65 enables near optimal performance, but a loss exceeding 85 results in little or no connectivity — thus incurring some maximum cost. Now, when the standard deviation of q is close to zero (path loss can be predicted with near certainty) UAVs can fly to positions where the expected path loss is 65, but still only incur a negligible penalty. However, as the standard deviation (and thus uncertainty) about q grows, the original piecewise linear function is smoothed such that, even when the expected path loss is < 65, UAV's incur a significant expected cost because of the increased risk associated with inaccurate predictions. This provides an incentive for UAVs to stay closer together in cases where signal loss cannot be predicted with certainty.

B. Integrating Sensing and Routing Decisions

We now discuss how, using max-sum, these connectivity costs can be integrated into UAV flight path decisions. Following Stranders *et al* (Sec. II-C), we achieve this by defining the ith UAV's individual action, a_i, as a flight path of length N starting from its current position; and define the *neighbourhood* joint action, $\mathbf{a}_i$, as the set comprising the individual actions of all UAVs (including i) within a fixed radius of i's position: $\mathbf{a}_i = \{a_j | j \in \mathcal{N}_i\}$. Now, in their original work, Stranders' *et al* define an agent's local utility as the reduction in entropy due to the observations made at positions visited by $\mathbf{a}_i$. Denoting this utility as $IG_i(\cdot)$, we now redefine the local utilty as

$$U_i(\mathbf{a}_i) = IG_i(\mathbf{a}_i) + \sum_{k=1}^{N} T_i^k(\mathbf{a}_i) \quad (4)$$

where $T_i^k(\cdot)$ is the *communication* cost incurred by UAV i while at the kth position on its flight path. Since this cost depends on the actions of all other UAVs that form part of i's link to the base station, one naïve way to compute this is to include in $\mathbf{a}_i$ the actions of all UAVs that could potentially route data between i and the base station. However, this would generally result in densely connected factor graphs that are computationally complex for max-sum to solve. Fortunately, many UAVs will share a large proportion of their routing tables — a fact that can be leveraged when calculating each $T_i^k(\mathbf{a}_i)$.

Specifically, we achieve this by passing an additional set of messages between UAVs in the following way. First, we denote the sequence of N positions visited by i while executing a_i as $m^1(a_i), \ldots, m^N(a_i)$, which we use to define the point-to-point channel cost between UAVs i and j at time k:

$$D_{i\leftrightarrow j}^k(a_i, a_j) = EL\left(m^k(a_i), m^k(a_j)\right) \quad (5)$$

In particular, we require that $D_{i\leftrightarrow i}^k(a_i, a_i) = 0$ and $\forall i \neq j$, $D_{i\leftrightarrow j}^k(a_i, a_j) < 0$, both of which may be expected of any reasonable cost function. Second, we define a new message type, $c_{i\to j}^k(a_i)$, which i sends to inform all $j \in \mathcal{N}_i$ of i's least cost communication path to the base station that does not go through j. Denoting the fixed position of the base station as p_b, this is defined as

$$\begin{aligned} c_{i\to i}^k(a_i) &= EL(m^k(a_i), p_b) & (6)\\ \forall_{i\neq j}\ c_{i\to j}^k(a_i) &= \max_{d\in\mathcal{N}_i\setminus j} D_{i\leftrightarrow d}^k(a_i, a_d^*) + c_{d\to i}^k(a_d^*) & (7)\end{aligned}$$

[5]Here, the distribution is truncated because the conditional probability for $q \notin \mathcal{Q}_i$ is 0.

where a_d^* is the best action for d, which UAV i can calculate by maximising the sum of its local factor and received messages:

$$\mathbf{a}_d^* = \arg\max_{a_d} \max_{\mathbf{a}_i \setminus \{i,d\}} U_i(\mathbf{a}_i) + \sum_{l \in \mathcal{N}_i} q_{l \to i}(a_l) \tag{8}$$

Thus, $c_{i \to j}^k(a_i)$ represents the total link cost (given a_i) for i's connection to the base station, assuming all other paths in $\mathbf{a}_i$ are chosen optimally. Based on this, $T_i^k(\cdot)$ can then be calculated by adding the corresponding point-to-point cost to each c_{i-d}^k and maximising over *all* $d \in \mathcal{N}_i$:[6]

$$T_i^k(\mathbf{a}_i) = \max_{d \in \mathcal{N}_i} D_{i \leftrightarrow d}^k(a_i, a_d) + c_{d \to i}^k(a_d) \tag{9}$$

In particular, when i's best option is to communicate directly with the base station itself, this equation yields the point-to-point channel cost between i and the base station:

$$T_i^k(\mathbf{a}_i) = D_{i \leftrightarrow i}^k(a_i, a_i) + c_{i \to i}^k(a_i) = EL(m^k(a_i), p_b) \tag{10}$$

As with the standard max-sum algorithm, these messages can be sent and updated asynchronously until they converge, or some other stopping condition is reached. Then, as before, the optimal flight path for each UAV is decided by choosing $a_i = \arg\max_{a_i} \sum_{j \in \mathcal{N}_i} q_{j \to i}(a_i)$, except that this now balances the expected entropy reduction against communication costs.

In addition to meeting this functional objective, the algorithm has two additional advantages. First, as a side effect of computing $T_i^k(\cdot)$, each UAV i knows that, to send data to the base station, it should forward packets to the neighbouring UAV that provides the least cost communication route (i.e. the UAV that provides the arg max of Eq. 9). Second, despite the addition of a new message type, the significant scaling properties of the max-sum algorithm are maintained. That is, the number of messages scales linearly with the number of edges in the factor graph (in this case, $\sum_i |\mathcal{N}_i|$) and computational cost of computing each message is $O(\prod_{j \in \mathcal{N}_i} |\mathcal{A}_j|)$. When the factor graph is sparse (i.e. most UAVs are not within direct range of each other)[7] this is significantly more scalable and efficient than solving the non-factored problem directly — which requires the full Cartesian product $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_m$, and is thus order $O(\prod_{j=1}^m |\mathcal{A}_j|)$. Moreover, the computational cost can be further reduced by implementing the action pruning techniques described in [3].

IV. Conclusions & Future Work

In this paper, we propose a UAV flight planning procedure that enhances the current state-of-the-art with two significant new features: (1) we propose an on-line learning approach that allows UAVs to adapt their behaviour to suit the radio propagation characteristics observed in their environment; and (2) we integrate the coordination algorithm introduced by [3] with a new message passing procedure for propagating information about network routing costs. Together, these features allow multiple UAVs to efficiently coordinate their flight paths and network routing strategies, so that each one maintains a reliable communication link with the base station while at the same time meeting their sensing objectives.

In future work, we plan to evaluate our approach by using both simulated and real quadrotor UAVs to find a missing person on the ground [4], [17]. Also, in our current approach, UAVs maintain a single multi-hop link with the base station. To encourage more robust operation, we plan investigate modifications to our algorithm that allow UAVs to weigh up the pros and cons of maintaining multiple connections, so that communication is not lost should one link fail.

Acknowledgment

This research is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) as part of the Sensing Unmanned Autonomous Aerial VEhicles (SUAAVE) project, grant reference EP/F06358X/1.

References

[1] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. Hedrick, "An overview of emerging results in cooperative uav control," in *the 43rd IEEE Conference on Decision and Control*, vol. 1, 2004, pp. 602–607.

[2] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, "Efficient planning of informative paths for multiple robots," in *Proceedings of IJCAI 2007*, 2007, pp. 2204–2211.

[3] R. Stranders, A. Farinelli, A. Rogers, and N. Jennings, "Decentralised coordination of mobile sensors using the max-sum algorithm," in *Proceedings of IJCAI 2009*, 2009, pp. 299–304.

[4] S. Cameron, S. Hailes, S. Julier, S. McClean, G. Parr, N. Trigoni, M. Ahmed, G. McPhillips, R. de Nardi, J. Nie, A. Symington, W. T. L. Teacy, and S. Waharte, "Suaave: Combining aerial robots and wireless networking," in *Proceedings of the 25th Bristol International UAV Systems Conference*, 2010.

[5] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1, pp. 111–131, 2008.

[6] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 165–172.

[7] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[8] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[9] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research*, vol. 9, pp. 2761–2801, 2008.

[10] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 42, no. 2, pp. 498–519, 2001.

[11] R. Stranders, A. Farinelli, A. Rogers, and N. J. Jennings, "Decentralised control of continously valued control parameters using the max-sum algorithm," in *Proceedings of AAMAS'09*, 2009, pp. 601–608.

[12] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 723–735, 2001.

[13] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings, "Decentralised coordination of low-power embedded devices using the max-sum algorithm," in *Proc. of AAMAS 2008*, 2008, pp. 639–646.

[14] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, February 2007.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[16] N. J. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*, 2nd ed. Wiley, 1994, vol. 1.

[17] W. T. L. Teacy, J. Nie, S. McClean, G. Parr, S. Hailes, S. Julier, N. Trigoni, and S. Cameron, "Collaborative sensing by unmanned aerial vehicles," in *Proceedings of ATSN'09*, 2009, pp. 13–16.

[6] Note that in Eq. 9, the values of a_i and a_d are specified in $\mathbf{a}_i$.

[7] Scalability can be further ensured by limiting $\mathcal{N}_i$ to the K nearest UAVs.